FIG. 1
PRIOR ART

```
1    c:\collections
2        notes.txt
3        myletter.doc
4        c-myhomepage
5
6          s
7            homepage.html
8            myphoto.jpg
```

FIG. 2

```
1    c:\collections
2        notes.txt
3        myletter.doc
```

100

```
4        c-myhomepage
5          cspec
6          s
7            homepage.html
8            myphoto.jpg
```

FIG. 3

102

```
1    collection        c-myhomepage
2    coll-type         cf-web-page
3    coll-desc         A sample homepage collection
4    end-collection
```

FIG. 4

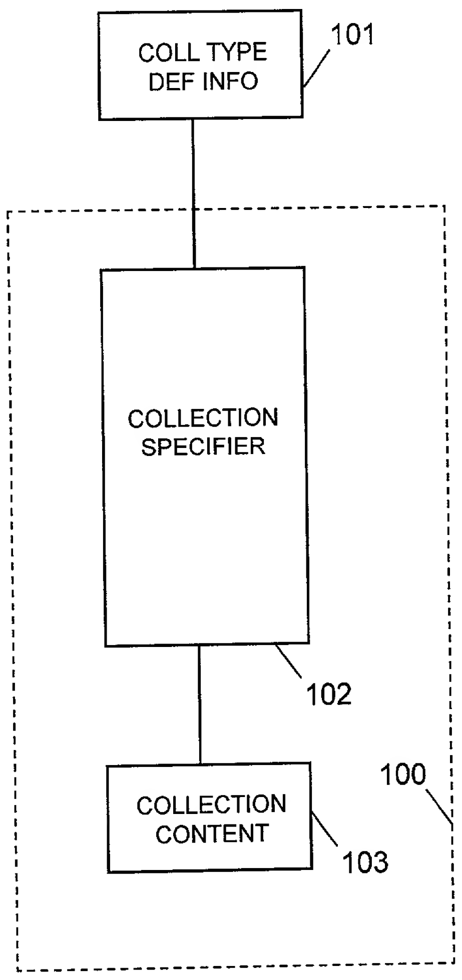

COLL TYPE DEF INFO
101
COLLECTION SPECIFIER
102
COLLECTION CONTENT
103
100

FIG. 5

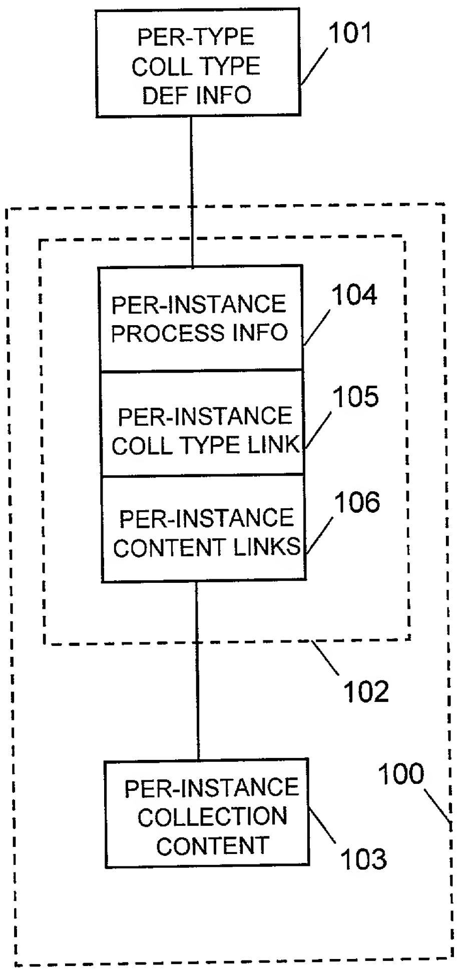

PER-TYPE COLL TYPE DEF INFO
101
PER-INSTANCE PROCESS INFO
104
PER-INSTANCE COLL TYPE LINK
105
PER-INSTANCE CONTENT LINKS
106
102
PER-INSTANCE COLLECTION CONTENT
103
100

FIG. 6

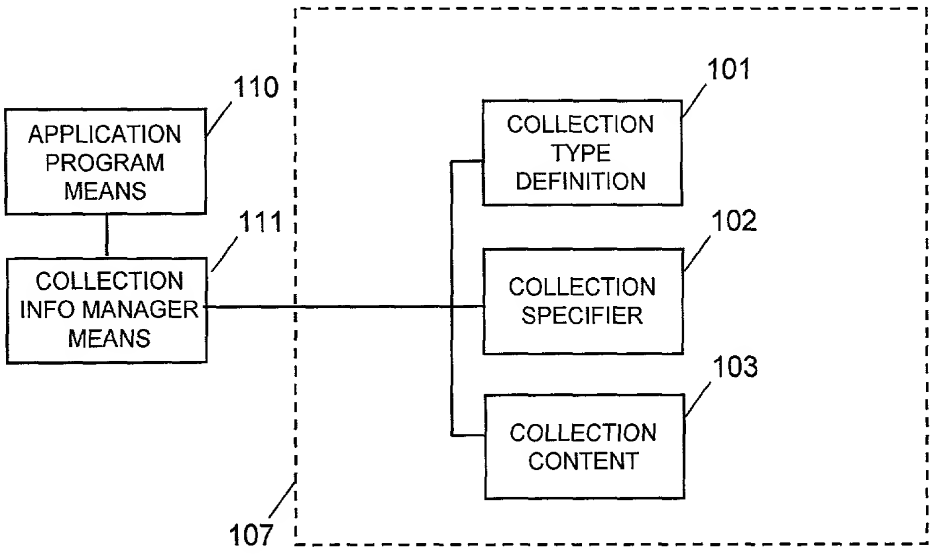

APPLICATION PROGRAM MEANS
110
COLLECTION INFO MANAGER MEANS
111
COLLECTION TYPE DEFINITION
101
COLLECTION SPECIFIER
102
COLLECTION CONTENT
103
107

FIG. 7

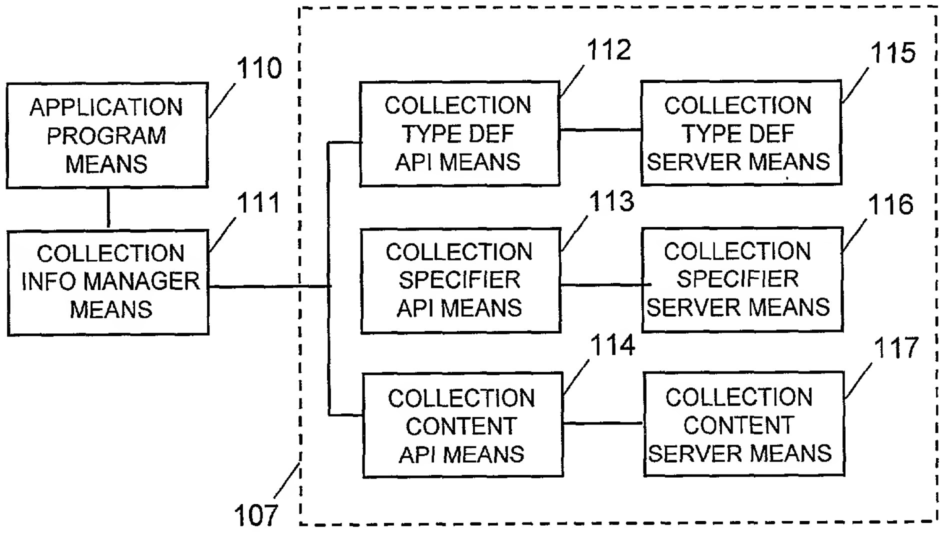

APPLICATION PROGRAM MEANS
110
COLLECTION INFO MANAGER MEANS
111
COLLECTION TYPE DEF API MEANS
112
COLLECTION TYPE DEF SERVER MEANS
115
COLLECTION SPECIFIER API MEANS
113
COLLECTION SPECIFIER SERVER MEANS
116
COLLECTION CONTENT API MEANS
114
COLLECTION CONTENT SERVER MEANS
117
107

FIG. 8

```
/* collection data structure */
collection-info {

  + specifier_info
      + coll-type-indicator
      + other specifier information ...

  + content_info
      + content_location_info ...
      + content_members ...
      + other content information...

  + other collection structure information...
}
```

FIG. 9

```
/* collection type definition data structure */
collection-type-definition-info {

  + coll-type-name
  + collection internal structure info ...
  + collection content location info ...
  + collection content type recognition info ...

  + other collection type definition information...
}
```

FIG. 10

```
     KEY                    VALUE

/* collection type internal structure definitions */
dir_source_files          ./s
dir_doc_files             ./doc

/* content location definitions (per-type content links) */
content_subtree_http      http://host.com/some/dir/name
content_subtree_ftp       ftp://host.com/some/dir/name
content_subtree_nfs       /some/local/directory/name

/* content type recognition definitions */
content_policy            subtree_below_cspec_file
content_file_type         .c       file_cpp
content_file_type         .c       file_c
content_file_type         .h       file_c_include
content_file_type         .doc     file_ms_word
content_file_type         .html    file_html
content_file_type         .xls     file_ms_excel

/* collection processing definitions */
compile_c_files           yes
compiler_windows          vc++
compiler_unix             gcc
build platforms           Win98, Win2000, gnulinux
process files             compile link
link libraries            stdio math sock

/* results dispatching definitions */
results_ftp_host          ftp.output.com
results_ftp_dir           c:\ftphome\collection\results
```

FIG. 11

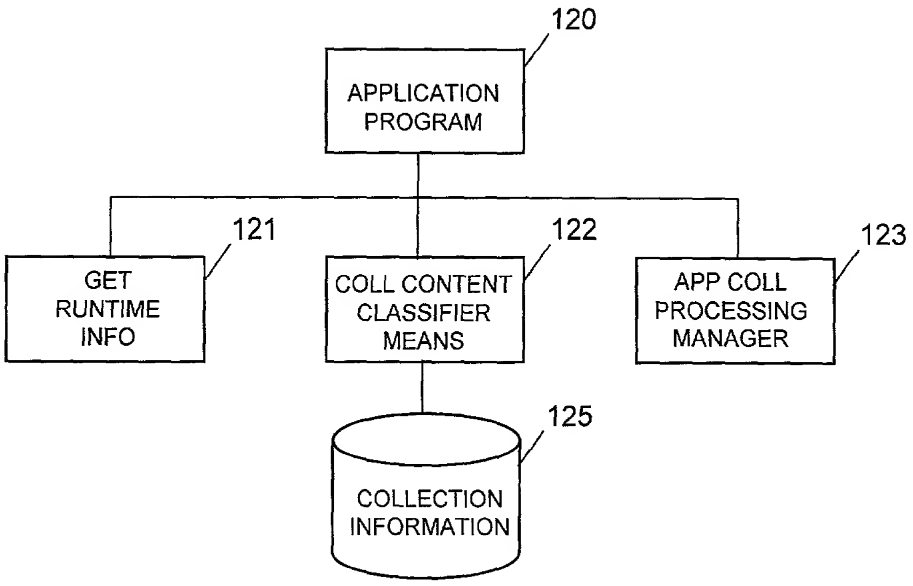
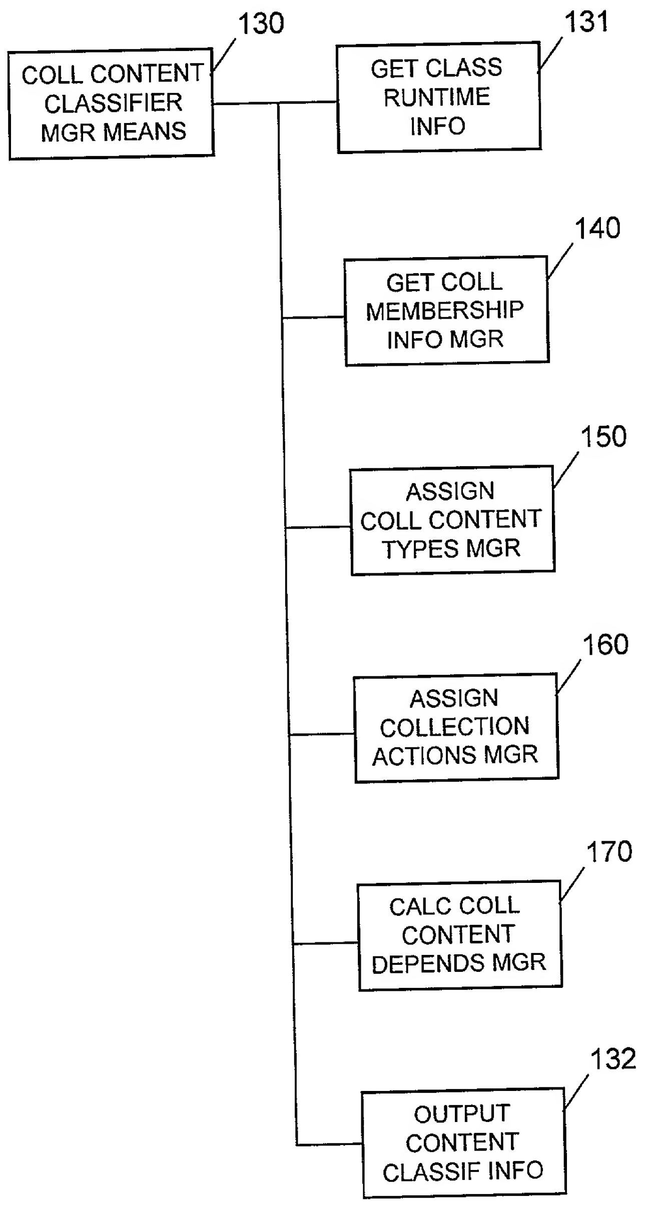


FIG. 12

```
1  /* simplified algorithm for classifier-enabled program*/
2  Call get runtime info to get collection to classify
3  Call collection content classifier to classify collection content
4  Call application processing functions to process the
     collection, using the content classification information
     produced in step 3.
```

FIG. 13



130
COLL CONTENT CLASSIFIER MGR MEANS
131
GET CLASS RUNTIME INFO
140
GET COLL MEMBERSHIP INFO MGR
150
ASSIGN COLL CONTENT TYPES MGR
160
ASSIGN COLLECTION ACTIONS MGR
170
CALC COLL CONTENT DEPENDS MGR
132
OUTPUT CONTENT CLASSIF INFO

FIG. 14

```
/* simplified algorithm for classifier manager module */
Build data structures
Obtain runtime info including a collection to classify

Obtain lists of collection products and product contents
Assign a content type to each content member
Assign a symbolic action to each content member
Calculate dependencies for each content member

Organize and output information
Return classifier data structures to caller
```

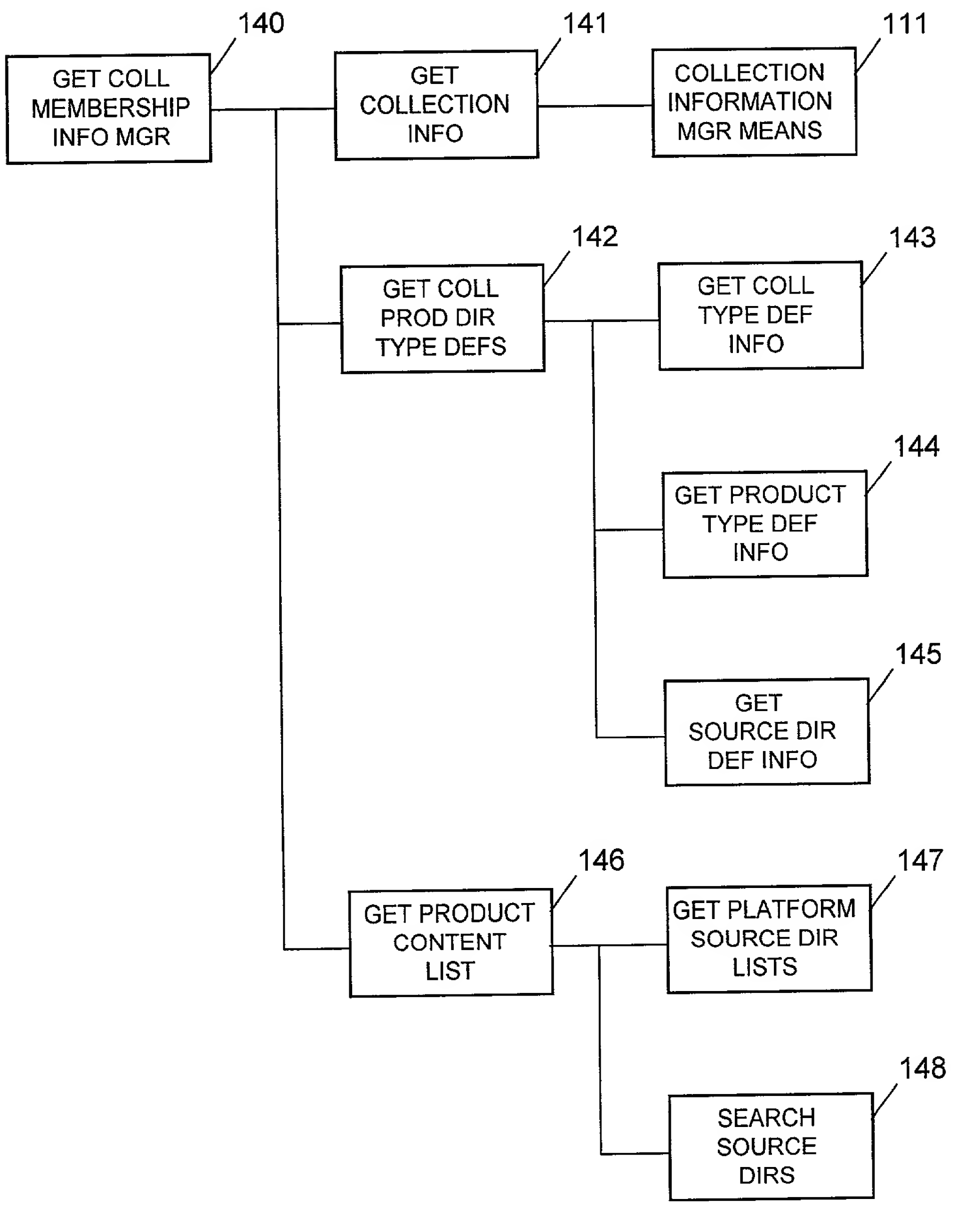
FIG. 15

```
/* top-level classified-content data structure */
coll-class-content {

/* classification info for this collection instance */
     + collection type indicator
     ... other collection info

/* classification info for this collection's products */
     + list of prod-class-content structures
         + list of content-class-entry structures

/* type definition info to support classification ops */
     + collection type definition information
         + product type definition information
             + file type identification information
             + content type definition information
                 + action type definition information

}
```

FIG. 16

```
1    collection type definition information
2      product type definition information
3        content type definition information
4          action type definition information


5    cspec:
6    coll-type                  ct-program

7    index-coll-types.tbl:
8    ct-program                 ct-program.def
9    ct-web-page                ct-web-page.def

10   ct-program.def:
11   product-type-index         index-product-types.tbl

12   index-product-types.tbl:
13   pt-program                 pt-program.def

14   pt-program.def:
15   content-type-index         index-content-types.tbl

16   index-content-types.tbl
17   content-c                  content-c.def

18   content-c.def:
19   action-type-index          index-action-types.tbl

20   index-action-types.tbl:
21   action-c                   action-c.def

22   action-c.def:
23   parser-type                internal
24   parser-name                internal-c
```

FIG. 17



140
GET COLL MEMBERSHIP INFO MGR
141
GET COLLECTION INFO
111
COLLECTION INFORMATION MGR MEANS
142
GET COLL PROD DIR TYPE DEFS
143
GET COLL TYPE DEF INFO
144
GET PRODUCT TYPE DEF INFO
145
GET SOURCE DIR DEF INFO
146
GET PRODUCT CONTENT LIST
147
GET PLATFORM SOURCE DIR LISTS
148
SEARCH SOURCE DIRS

FIG. 18

```
/* simplified algorithm for get coll membership info manager */
Build data structures

/* Obtain collection specifier info including coll product list */
Call Get Collection Specifier Info
    Call Collection Information Manager

/* Obtain type definition info for coll type, product type, etc */
Call Get Collection Product Dir Type Information
    Call Get Collection Type Definition Information
    Call Get Product Type Definition Information
    Call Get Product Source Directory Definition Information

/* Build per-product content lists from per-product src dir lists */
For each product:
  Call Get Platform Source Directory List
  Call Search Source Dirs to find content files in the source dirs
  Add found content files to content-entry data structures

Return completed prod-class-content list data struct to caller
```

FIG. 19

```
/* prod-class-content data structure for one product  */
prod-class-content {

    + product type indicator
    + list of product actions
    + list of content-class-entry data structures

    ... other product information
}
```

FIG. 20

```
/* content-class-entry data struct  for one content entry */
content-class-entry {

    + content entry pathname
    + content type indicator
    + content language type indicator
    + content action indicator
    + list of content dependencies
    + ...

}
```

FIG. 21

```
index-coll-types.tbl:
ct-program                  ct-program.def
ct-library                  ct-library.def
ct-doc-html                 ct-html.def

ct-program.def:
/* type definition info for a “ct-program” collection type */
product-type-index          index-prod-program.tbl
action-type-index           index-coll-action-types.tbl
action                      cleanup
action                      checkin
... other collection type info
```

FIG. 22

```
index-prod-program.tbl:
pt-program                  pt-program.def
pt-program-java             pt-program-java.def
pt-program-unix             pt-program-unix.def
pt-program-win              pt-program-win.def

pt-program.def:
/* type definition info for a “program” product type */
dir-source-files            source-dirs.lst
dir-library-files           library-dirs.lst
file-identification-table   file-identification.tbl
content-type-index          index-content-types.tbl
action-type-index           index-prod-action-types.tbl
action                      link
... other product type info
```

FIG. 23

```
source-dirs.lst:

dir/gnulinux2       ../s/gnulinux2
dir/gnulinux2       ../s/gnulinux
dir/gnulinux2       ../s/unix
dir/gnulinux2       ../s/pi
dir/gnulinux2       ../s

dir/win98           ../s/win98
dir/win98           ../s/win
dir/win98           ../s/pi
dir/win98           ../s

dir/win95           ../s/win95
dir/win95           ../s/win
dir/win95           ../s/pi
dir/win95           ../s
```

FIG. 24

```
library-dirs.lst:

dir/gnulinux2       ../lib/gnulinux2
dir/gnulinux2       ../lib/gnulinux
dir/gnulinux2       ../lib/unix
dir/gnulinux2       ../lib/pi
dir/gnulinux2       ../lib

dir/win98           ../lib/win98
dir/win98           ../lib/win
dir/win98           ../lib/pi
dir/win98           ../lib
```

FIG. 25

```
1    c:\collections
2        c-mystuff
3            cspec

4            s
5                pi
6                    file1.h
7                    file1.c
8                win
9                    file2-pd.h
10                   file2-pd.c
11               win98
12                   file2-pd.c
13               unix
14                   file2-pd.h
15                   file2-pd.c
16               gnulinux2
17                   file2-pd.c

18           lib
19               pi
20                   libfuns.c
21                   libfuns2.c
22               win
23                   libfuns.h
24                   libfuns2.c
25               unix
26                   libfuns.h

27           win98.plt
28               … compiled files, programs, libs, etc

29           gnulinux2.plt
30               … compiled files, programs, libs, etc
```

FIG. 26

```
/* file list for win98 platform */
../s/win98/file2-pd.c
../s/win/file2-pd.h
../s/pi/file1.h
../s/pi/file1.c
../lib/win/libfuns.h
../lib/win/libfuns2.c
../lib/pi/libfuns.c


/* file list for gnulinux2 platform */
../s/gnulinux2/file2-pd.c
../s/unix/file2-pd.h
../s/pi/file1.h
../s/pi/file1.c
../lib/unix/libfuns.h
../lib/unix/libfuns2.c
../lib/pi/libfuns.c
```

FIG. 27

```
collection        c-my-example
coll-type         ct-program
coll-desc         A memdir and memfile example.
end-collection

product           myprog
prod-type         pt-program
memdir            c:\third-party\source-files
memdir-prepend    c:\search\this\dir\first
memdir-append     c:\search\this\dir\last
memfile           c:\third-party\product\somefile.obj
memfile-prepend   c:\somewhere\file.c
memfile-append    c:\somewhere\file2.c
end-product
```

FIG. 28

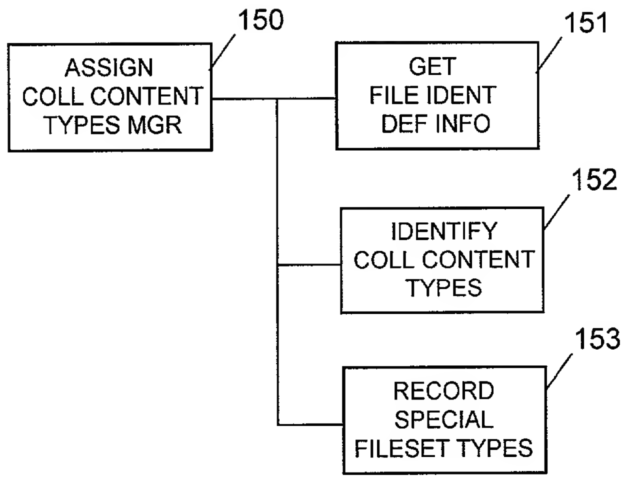

150
ASSIGN COLL CONTENT TYPES MGR
151
GET FILE IDENT DEF INFO
152
IDENTIFY COLL CONTENT TYPES
153
RECORD SPECIAL FILESET TYPES

FIG. 29

```
/* simplified algorithm for assign content type manager */
Build data structures

/* Obtain file identification criteria to enable matching */
Call Get File Identification Definition Information

/* Identify the content types of all files in the content list */
For each product in collection product list:
        For each file in current product content list:
                Call Identify Content Type to assign a content type

/* Record special fileset types */
Call Record Special Fileset Types

Return data structures to caller
```

FIG. 30

```
1    file-identification.tbl:
2    /* Match-type     Match-string     Content-type */

3    exact             junkfile         ignore
4    suffix            .bak             ignore
5    tail              ~                ignore
6    content           #!/bin/csh       csh
7    suffix            .c               c-source
8    suffix            .h               c-header
9    suffix            .C               c++
10   suffix            .c++             c++
```

FIG. 31

```
1    collection        c-my-example
2    coll-type         ct-program
3    coll-desc         A fileset example.
4    end-collection

5    product           myprog
6    prod-type         pt-program

7    fileset           myset ../s/file1.f  ../s/file2.f
8    fileset           myset ../s/file3.f
9    fileset-type      myset content-no-optimize

10   end-product
```

FIG. 32

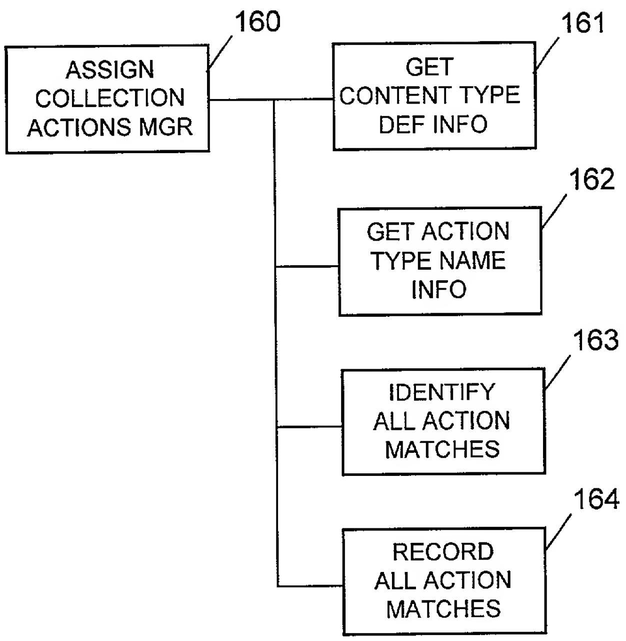

160
ASSIGN COLLECTION ACTIONS MGR
161
GET CONTENT TYPE DEF INFO
162
GET ACTION TYPE NAME INFO
163
IDENTIFY ALL ACTION MATCHES
164
RECORD ALL ACTION MATCHES

FIG. 33

```
1  /* simplified algorithm for assign content actions manager */
2  Build data structures

3  /* Obtain file type definition and action name index info */
4  Get File Type Definition Information
5  Get Action Type Name Information

6  /* Assign action names to collection, product, and content entries */
7  Call Identify All Action Matches to choose coll, prod, content actions
8  Call Record All Action Matches to record choices in data structures

9  Return completed data structures to caller
```

FIG. 34

```
index-content-types.tbl:
c-source                 content-c.def
c-header                 content-c-h.def
csh                      content-csh.def
html                     content-html.def

content-c.def:
/* type definition info for a "c" file type */
type                     c-source
language                 c
action                   action-c-source
action-type-index        index-action-types.tbl
... other content type definition info
```

FIG. 35

```
index-action-types.tbl:
action-c-source          action-c-source.def
action-c-header          action-c-header.def
action-csh               action-csh.def
action-html              action-html.def

action-c-source.def:
parser-type              internal
parser-name              internal-c
```

FIG. 36

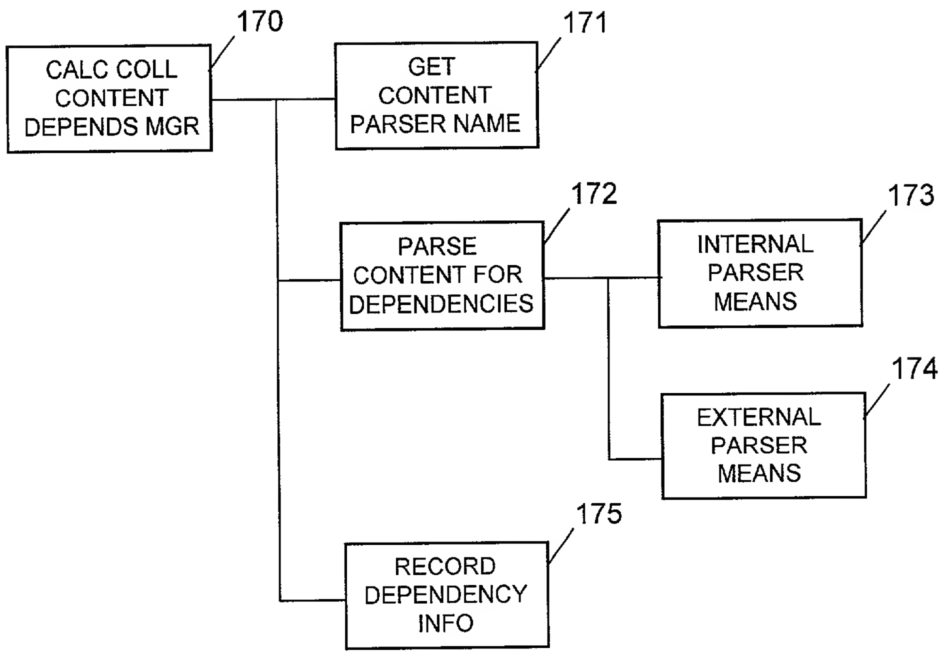


FIG. 37

```
1  /* simplified algorithm for calc content dependencies mgr */
2  Build data structures

3  /* loop over and parse all content entries */
4  For each product in collection product list:
5      For each file in current product content list:
6          Call Get Content Parser to choose a parser
7          Call Parse For Dependencies to obtain dependencies
8          Call Record Dependency Info to store dependencies in
                content-class-entry data structure

9  Return completed content-class-entry data structures to caller
```

FIG. 38

```
/* external parser invocation command line */
new-language-parser  infile outfile
new-language-parser  file1.c  file1.c.deps
```

```
file1.c.deps:
/* dependency info from new-language-parser */
dep    file1.c         stdio.h
dep    file1.c         app-data.h
dep    file1.c         content-class-entry.h
```

FIG. 39

```
c:\collections
    c-my-example
        cspec
        s
            pi
                cmdline.h
            win98
                cmdline.c
            gnulinux2
                cmdline.c
        lib
            pi
                libfuns.h
                libfuns.c
```

FIG. 40

```
cspec:

collection    c-my-example
coll-type     ct-program
coll-desc     A multi-platform C program with library.
end-collection

product       myprog
prod-type     pt-program
prod-desc     A program product.
end-product

product       mylibrary
prod-type     pt-library
prod-desc     A library product.
end-product
```

FIG. 41

```
/* classification output for win98 platform */
collection            c-my-example
coll-type             ct-program
coll-action           cleanup
coll-action           checkin
... other coll class info


/* classification info for a program product */
product               myprog
prod-type             pt-program
prod-action           link

content               cmdline.h
content-path          ../s/pi/cmdline.h
content-type          c-header
content-language      c
content-action        none
end-content

content               cmdline.c
content-path          ../s/win/cmdline.c
content-type          c-source
content-language      c
content-action        action-c-source
content-dep           ../s/pi/cmdline.h
content-dep           ../lib/pi/libfuns.h
end-content

end-product
```

FIG. 42

```
/* classification output */
collection            c-my-example

... /* classification info for the host collection */
... /* classification info for the program product */

/* classification info for a library product */
product               mylibrary
prod-type             pt-library
prod-action           archive
prod-action           export-lib-for-sharing

content               libfuns.h
content-path          ../lib/pi/libfuns.h
content-type          c-header
content-language      c
content-action        export-incl-for-sharing
end-content

content               libfuns.c
content-path          ../lib/pi/cmdline.c
content-type          c-source
content-language      c
content-action        action-c-source
content-dep           ../lib/pi/libfuns.h
end-content

end-product
```